A Three-regime Model of Network Pruning

Yefan Zhou [1 2] **Yaoqing Yang** [3] **Arin Chang** [2] **Michael W. Mahoney** [1 2 4]

Abstract

Recent work has highlighted the complex influence training hyperparameters, e.g., the number of training epochs, can have on the prunability of machine learning models. Perhaps surprisingly, a systematic approach to predict precisely how adjusting a specific hyperparameter will affect prunability remains elusive. To address this gap, we introduce a phenomenological model grounded in the statistical mechanics of learning. Our approach uses *temperature-like* and *load-like* parameters to model the impact of neural network (NN) training hyperparameters on pruning performance. A key empirical result we identify is a sharp transition phenomenon: depending on the value of a load-like parameter in the pruned model, increasing the value of a temperature-like parameter in the pre-pruned model may either enhance or impair subsequent pruning performance. Based on this transition, we build a three-regime model by taxonomizing the *global structure* of the pruned NN loss landscape. Our model reveals that the dichotomous effect of high temperature is associated with transitions between distinct types of global structures in the post-pruned model. Based on our results, we present three case-studies: 1) determining whether to increase or decrease a hyperparameter for improved pruning; 2) selecting the best model to prune from a family of models; and 3) tuning the hyperparameter of the Sharpness Aware Minimization method for better pruning performance.

1. Introduction

A recently-popular approach to compressing large neural networks (NNs) is to perform pruning, i.e., to remove unnecessary weights from a trained model. The resulting "sparser" NNs often have improved memory and inference efficiencies, compared to the original "denser" NNs.

A common approach to pruning (Lecun et al., 1989; Han et al., 2015; Molchanov et al., 2017) involves adopting a three-stage train-prune-retrain pipeline: 1) train a large or over-parameterized dense model to some sort of convergence; 2) prune the dense model to obtain a sparse model; and then 3) retrain the sparse model to recover its performance. A considerable amount of work has focused on improving the sub-network performance in the (second) pruning stage (Blalock et al., 2020) and the (third) retraining stage (Renda et al., 2020; Le & Hua, 2021). However, there remains little guidance for the (first) stage of dense model training, i.e., how to improve the prunability of the original large model (Rosenfeld et al., 2021). Recent work has shown that tuning optimization-related hyperparameters, such as the number of training epochs (Li et al., 2020; Liu et al., 2021; Shen et al., 2022) and the batch size (Barsbey et al., 2021), can potentially benefit specific pruning methods. Despite these findings, a principled approach to predict when and how adjusting a given hyperparameter during the stage of dense model training will impact subsequent pruning performance remains to be developed.

In this paper, we develop a simple operational model for NN pruning, focusing on the optimal selection of hyperparameters, such as the number of training epochs, in the (first) training stage. Our model is inspired by recent work in the statistical mechanics of learning (Yang et al., 2021; Martin & Mahoney, 2017; 2021b). In the statistical mechanics approach to learning, multiple qualitatively different "phases" of behavior can arise in black-box Deep NNs. This concept is explicitly illustrated by the Very Simple Deep Learning (VSDL) model, proposed in Martin & Mahoney (2017). Within the VSDL model, these phases and the sharp transitions between them can be identified on a two-dimensional "phase" diagram, where the x and y axes represent load-like and temperature-like parameters, respectively. Load-like parameters characterize the relationship between the quantity and/or quality of data, relative to model size. This was represented as label noise in training data by Martin & Mahoney (2017) and as width scaling by Yang et al. (2021). Temperature-like parameters, on the other hand, characterize the magnitude of noise introduced during stochastic training. This can be represented in terms of common hyper-

[1]International Computer Science Institute, CA, USA [2]University of California, Berkeley, CA, USA [3]Dartmouth College, NH, USA [4]Lawrence Berkeley National Laboratory, CA, USA. Correspondence to: Yefan Zhou <yefan0726@berkeley.edu>.

Proceedings of the 40th International Conference on Machine Learning, Honolulu, Hawaii, USA. PMLR 202, 2023. Copyright 2023 by the author(s).

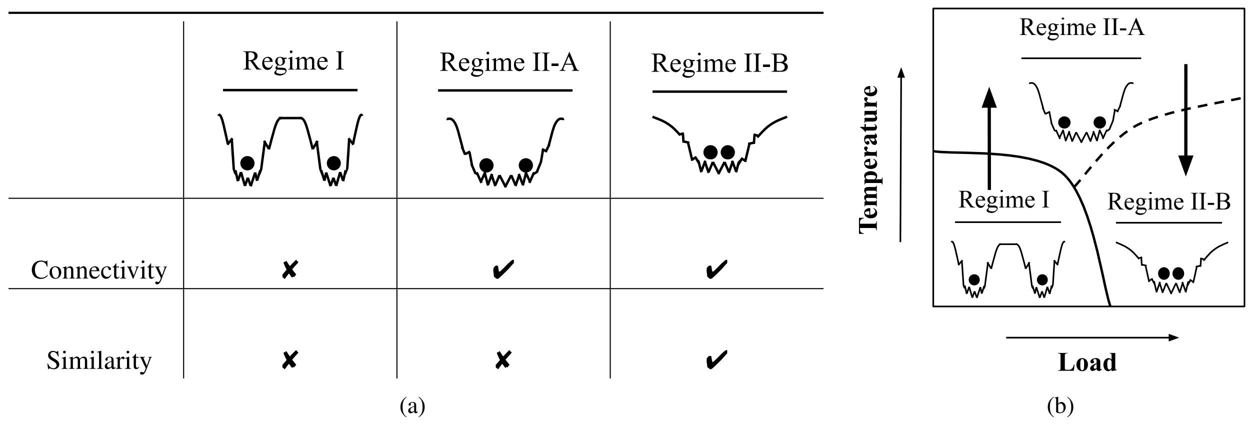


Figure 1. The three regimes of pruning obtained by varying temperature-like parameters (in the dense pre-pruned model) and load-like parameters (in the sparse post-pruned model): loss landscape connectivity metrics such as LMC identify Regime I versus Regime II; and loss landscape similarity metrics of outputs between models in well-connected regimes then identify Regimes II-A and II-B. The regimes are thus Regime I (poorly-connected loss landscapes); Regime II-A (well-connected but relatively dissimilar model outputs); and Regime II-B (well-connected and relatively similar outputs). For a given load goal (density of the pruned model), we focus on the favorable transitions from Regime I to Regime II-A (obtained by increasing the temperature) and from Regime II-A to Regime II-B (obtained by decreasing the temperature), as indicated by the arrows.

parameters for training, including the number of epochs and batch size in Martin & Mahoney (2017); Yang et al. (2021). Using the notions of load-like and temperature-like parameters, Yang et al. (2021) provided a comprehensive taxonomy of NN loss landscapes, employing easily computed metrics to identify, predict, and distinguish qualitatively different phases of model training.

Our three-regime VSDL model for pruning. We construct a novel model for NN pruning that is based on the VSDL model. In our model, the load-like parameter is represented as a distinct hyperparameter used in the pruning process, namely the density of the pruned models; and the temperature-like parameter is represented as common hyperparameters for the dense model training (e.g., training epochs, batch size), similar to Martin & Mahoney (2017); Yang et al. (2021).

Our empirical results validate the effectiveness of the proposed model, demonstrating that adjusting the load and temperature parameters can lead to relatively-sharp transitions in model performance and that making decisions based on this leads to improved test error for pruned models. Moreover, our work confirms the metrics previously used to develop the taxonomy of loss landscapes (Yang et al., 2021), showcasing their applicability to the very different problem of model pruning. Ultimately, our results contribute to the more efficient, metric-informed selection of temperature-like hyperparameters (depending on the load), offering practical approaches for applying VSDL models.

In more detail, we consider two metrics to measure NN loss landscapes, namely the linear mode connectivity (LMC) (Garipov et al., 2018; Draxler et al., 2018; Frankle et al., 2020) and the centered kernel alignment (CKA) similarity (Kornblith et al., 2019). LMC quantifies how well different local minima are connected to each other in the loss landscape, and thus it captures the *connectivity* between trained models. CKA similarity, on the other hand, is used to capture the *similarity* between the outputs of models. Yang et al. (2021) used these two metrics to measure the *global structure* of loss landscapes. Here, we apply these insights to the model pruning problem. As indicated in Figure 1, our results empirically produce a three-regime taxonomy based on the similarity and connectivity of pruned models' loss landscape, effectively diagnosing and explaining the pruning performance as load-like and temperature-like control parameters are varied.

Our main contributions are as follows:

- We construct a VSDL model to study the different types of model training in order to improve pruning performance. Our model is taxonomized into three regimes based on the connectivity and similarity of the pruned model loss landscape. This is graphically represented in Figure 1.
- Our three-regime model effectively identifies and predicts a dichotomous phenomenon: depending on the value of the target load parameter and the measured values of the LMC and CKA, one can obtain improved pruned models either by increasing or decreasing the temperature parameter. Our three-regime model demonstrates that the phenomenon is well correlated with the transition among the three regimes, as shown in Figure 1b.

- Our new insights on three-regime pruning lead to new practical approaches to improving pruning, which we present as three case-studies: 1) given initial load and temperature hyperparameters, we can predict the correct direction to adjust the temperature in order to achieve optimal pruning performance; 2) for a given target model load, we can design a new model selection method based on connectivity and similarity that can predict the optimal temperature parameter without the need for an exhaustive grid search; and 3) we find that training models with the Sharpness Aware Minimization (SAM) method results in improved pruning when the temperature-like parameter of SAM is optimally tuned using our three-regime model.

Our code is open-sourced.[1] We provide a comprehensive overview of related work as well as provide further discussion of two aspects of our approach that differ from how Yang et al. (2021) applied the VSDL model in Appendix A.

2. Background and Setup

In this section, we provide background and a basic setup for our main results.

2.1. Load and temperature

Load and temperature parameters enter naturally when considering the statistical mechanics approach to learning and generalization, both historically (Seung et al., 1992; Watkin et al., 1993; Haussler et al., 1996; Engel & den Broeck, 2001), where they enter directly, as well as more recently (Martin & Mahoney, 2017; 2021b), where they enter indirectly via other control parameters.

Informally, a *load-like parameter*, which we will denote generically by L, refers to some quantity related to the quantity/quality of the data, relative to the size of the model. Here, we mainly study the models after pruning, and we vary the model density L_{density} to change the load. In this case, decreasing L_{density} corresponds to having more data, relative to the model size. We also study alternative load parameters, e.g., model width scaling L_{width} or depth L_{depth} of the pruned model, given a fixed model density.

Similarly, a *temperature-like parameter*, which we will denote generically by T, refers to some quantity related to the empirical noise/stochasticity introduced in the learning process. We mainly adjust the temperature by varying the number T_{epoch} of training epochs (in the first stage of the train-prune-retrain pipeline) or the batch size T_{batch}. The neighborhood size ρ, a hyperparameter in SAM (that indicates the magnitude of the adversarial perturbations over the model parameters before each gradient update of SGD optimizer), can also be viewed as a temperature-like parameter (as we show in Section 4.3). In practice, an increase in temperature corresponds to fewer training epochs, smaller batch size, and larger neighborhood size.

2.2. Preliminaries

Consider a NN $f(\mathbf{x};\Theta) : \mathbb{R}^{d_{\text{in}}} \to \mathbb{R}^{d_{\text{out}}}$ with input $\mathbf{x}$ and parameters $\Theta \in \mathbb{R}^m$. We train the NN using mini-batch SGD, with batch size T_{batch}, and total number of epochs T_{epoch}. For randomly initialized parameters, Θ_0, SGD randomness, ξ, the process of training with specific temperature-like parameters is denoted by $\text{Train}(\Theta_0, \xi, T)$, $T = \{T_{\text{epoch}}, T_{\text{batch}}, ...\}$. We mainly study NNs in classification tasks, using $\text{error}_{\text{train}}(\Theta)$ and $\text{error}_{\text{test}}(\Theta)$ to represent the error on the training and test sets, respectively.

2.3. Network pruning

We mainly investigate unstructured pruning, which removes weights without considering model structure.

Model density. We define the model density, denoted as $L_{\text{density}} = \frac{|\mathcal{M}|}{m}$, as the fraction of the number of parameters preserved to the total parameter count, where $\mathcal{M} \in \{0,1\}^N$ is the binary pruning mask informing the position of remaining weights. A pruned model is denoted as $\Theta \odot \mathcal{M}$, where $\odot$ indicates the element-wise product. The process of pruning a model with specific width/depth into a target density is denoted by $\text{Prune}(\Theta, L)$, $L = \{L_{\text{density}}, L_{\text{width}}, L_{\text{depth}}, ...\}$.

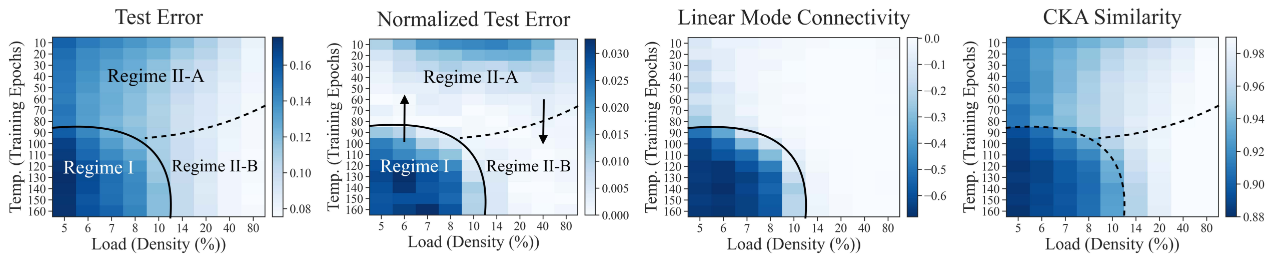
Pruning strategies. Magnitude pruning (MP) utilizes the absolute value of each parameter as a measure of importance to determine the mask $\mathcal{M}$. Unstructured MP retains the top-L_{density} percent of important parameters, with two variants based on the layer-wise distribution of density: 1) UNIFORMMP, which prunes each layer uniformly, and 2) GLOBALMP, which imposes a global threshold on parameter magnitudes of every layer for achieving the desired global density target.

2.4. Loss landscape metrics

Recent work (Yang et al., 2021) proposed an extensive taxonomy of the loss landscape of realistic NNs, finding (among other things) that LMC and CKA similarity can be used to determine the phase in which a trained model lies. In this study, we leverage these metrics to pinpoint the particular regime within our three-regime model to which a pruned NN belongs.

LMC. Frankle et al. (2020) shows that linear low-loss paths can be found between two networks if they originate from shared trained initialization, motivating us to take the linear variant of the mode connectivity used in Yang et al. (2021). Given two separate sets of weights Θ, Θ', we parameterize

[1] https://github.com/YefanZhou/ThreeRegimePruning



(a) Test error (b) Normalized Test error (c) LMC (d) CKA Similarity

Figure 2. Partitioning the 2D model density (load) – training epoch (temperature) diagram into three regimes. Models are trained with PreResNet-20 on CIFAR-10. The y-axis denotes a temperature-like parameter, indicated by a range of training epochs preceding the pruning process, while the x-axis represents a load-like parameter, expressed through diverse model densities applied to the model. (a) Final test error of the models after pruning and retraining. (b) Normalized test error is obtained by subtracting the optimal (lowest) test error from each column of the diagram in (a). The black arrows indicate two favorable transitions to lower test error regimes given a fixed model density. (c) LMC forms a sharper boundary that distinguishes Regime I from Regime II. (d) CKA shows a smooth transition that categorizes Regimes II-A and II-B.

the linear path $\gamma(t), t \in [0,1]$ connecting them with $\gamma(t) = t\Theta + (1-t)\Theta'$. Then, the LMC is defined as

$$\texttt{lmc}(\Theta,\Theta') = \frac{1}{2}\left(\text{error}_{\text{train}}(\Theta) + \text{error}_{\text{train}}(\Theta')\right) - \text{error}_{\text{train}}\left(\gamma(t^*)\right), \tag{1}$$

where $\gamma(t) = t\Theta + (1-t)\Theta'$ and t^* maximizes $t \mapsto \left|\frac{1}{2}\left(\text{error}_{\text{train}}(\Theta) + \text{error}_{\text{train}}(\Theta')\right) - \text{error}_{\text{train}}(\gamma(t))\right|$, $t \in [0,1]$.

We consider two cases for LMC. If $\texttt{lmc}(\Theta,\Theta') \approx 0$, then this implies a curve of low training error connecting Θ, Θ', rendering the loss landscape well-connected. If $\texttt{lmc}(\Theta,\Theta') < 0$, then this means that there is a "barrier" of high training error between Θ and Θ'. In this case, we say that the loss landscape is poorly connected (or that the LMC is poor).[2]

CKA similarity. CKA similarity measures the representational similarity of two parameter configurations Θ, Θ' in the output space, as measured by the centered kernel alignment (CKA) metric. Following Yang et al. (2021), we measure the distance between two models using their predictions, instead of weights, to avoid having low similarity resulting from different weights, even though the predictive functions are similar. Let $\{\mathbf{x}_1, \ldots, \mathbf{x}_s\}$ denote a set of s randomly sampled datapoints, and $F_\Theta = \begin{bmatrix} f(\mathbf{x}_1;\Theta) & \cdots & f(\mathbf{x}_s;\Theta) \end{bmatrix}^\top \in \mathbb{R}^{m \times d_{\text{out}}}$ denote the concatenation of the outputs of the network. In this case, the CKA similarity between the two weights Θ and Θ' is given by

$$\text{cka}(\Theta,\Theta') = \frac{\text{Cov}(F_\Theta, F_{\Theta'})}{\sqrt{\text{Cov}(F_\Theta, F_\Theta)\,\text{Cov}(F_{\Theta'}, F_{\Theta'})}}, \tag{2}$$

where $\text{Cov}(X,Y) = (m-1)^{-2}\,\text{tr}\left(XX^\top H_m YY^\top H_m\right)$, for $X, Y \in \mathbb{R}^{s\times d}$, and $H_m = I_m - m^{-1}\mathbf{1}\mathbf{1}^\top$ is a centering matrix.

Note that measuring the LMC and the CKA similarity defined above requires two distinct weights Θ and Θ' trained on the same sparse architecture. To obtain the two weights, we first prune a trained full dense model, and we then retrain two copies of the pruned sparse model with different SGD noise for α epochs. Formally, these two weights are defined as $\Theta = \text{Train}\left(\Theta \odot \mathcal{M}, \xi^1, T_{\text{epoch}}\right)$ and $\Theta' = \text{Train}\left(\Theta \odot \mathcal{M}, \xi^2, T_{\text{epoch}}\right)$, $T_{\text{epoch}} = \alpha$.

3. Validity of the Three-regime Model

In this section, we report our empirical results, illustrating the validity of the three-regime model we introduce.

3.1. Experimental setup

We generate thousands of pruned models by systematically varying the load and temperature parameters. A variant of residual networks with 20 layers (PreResNet-20) was trained on CIFAR-10 with various temperature hyperparameters and pruned to different load (density) levels.

We tune the following hyperparameters to vary the magnitude of load parameters and temperature parameters:

- Model density (Load): we consider the model pruned to 9 densities: $\{5, 6, 7, 8, 10, 14, 20, 40, 80\}\%$.
- Number of training epochs (Temperature): we consider training to numbers of epochs that are multiples of 10 $\{10, 20, 30, \ldots, 160\}$. Training for a total of 160 epochs corresponds to no early stopping.
- Batch size (Temperature): we consider training the

[2]One difference from Yang et al. (2021) is that we study the "linear version" of mode connectivity (Frankle et al., 2020), which provides faster computation and is easier for downstream tasks, as discussed in Section 4. We also note that Yang et al. (2021) discuss cases when the mode connectivity is larger than 0, which happens rarely when we use the linear version of mode connectivity.

model with varying batch sizes {16, 21, 27, 32, 38, 44, 52, 64, 92, 128, 180, 256, 512} while keeping the same amount of training iterations.

We study the UNIFORMMP as the basic pruning method. More details on the experimental setup can be found in Appendix B.1.

3.2. Regimes of loss landscape

We summarize the results in Figure 2. Each pixel within the sub-figures represents a unique experimental configuration, where a model is trained for a specific number of epochs (y-axis) and then pruned to a particular density (x-axis). Figure 2a displays the test error of each model after pruning and retraining, while Figure 2b showcases the test error normalized using a scheme applied to each model density (column). This normalization process involves subtracting the test error by the optimal early-stopped test error for each density-related column. Additionally, Figure 2c and 2d present the LMC and CKA of the pruned models, respectively. The normalization scheme applied to Figure 2a results in Figure 2b, enabling the comparison of performance differences across various regimes at each density level. Notably, Figure 2b demonstrates that Regime II-A typically surpasses Regime I in performance but falls short when compared to Regime II-B.

We observe that the connectivity and similarity metrics can be used to find the transitions observed across the model density (load) – training epochs (temperature) phase space, forming a three-regime classification that classifies the loss landscape, as shown in Figure 1.

- **LMC distinguishes models with poorly connected loss landscape, which we categorize as Regime I**. The first transition is displayed in Figure 2c. The white region represents the near-zero LMC, which implies a flat curve in the loss landscape between two local minima; and the dark blue region represents the negative LMC, which implies a high loss barrier moving from one local minimum to another. The transition in LMC forms a curve separating Regime I from Regime II.
- **CKA similarity further categorizes models with well-connected loss landscapes into models with similar/dissimilar outputs.** In Figure 2d, CKA divides the region where LMC is near-zero in Figure 2c into Regime II-A and Regime II-B, the latter including models with more similarity.[3] Regime I observes a much smaller CKA similarity than the other two regimes, which coincides with negative LMC. That the CKA similarity diagram exhibits a smooth transition between regimes, rather than a sharp transition from a negative value to zero, as is observed in LMC, is consistent with results in Yang et al. (2021).
- **To enumerate the three regimes**:
  – **Regime I:** Poorly connected loss landscape, less similar output representations.
  – **Regime II-A:** Well-connected loss landscape, less similar output representations.
  – **Regime II-B:** Well-connected loss landscape, relatively large output similarity.

[3]The transition between Regime II-A and Regime II-B is much smoother, and it need not be viewed as a finite-sized approximation to a "phase transition," in contrast to the much sharper transition between Regime I from Regime II that LMC identifies.

Based on the three-regime taxonomy, we assert the following as the central claim of this work.

Main claim. Given a model, we can use the loss landscape metrics to inform the optimal temperature to apply in the training stage at each possible level of its load (model density) after pruning.

To elaborate further, we mainly use the connectivity and similarity of the loss landscape, measured by LMC and CKA similarity respectively. Low-density models with poor connectivity benefit from increased temperature, facilitating a transition from Regime I to Regime II-A. Conversely, high-density models with good connectivity benefit from decreased temperature, enabling a transition from Regime II-A to Regime II-B to improve similarity.

3.3. Corroborating results

We have additional results that modify the basic setup of Section 3.2. These are described in more detail in Appendix B.2. In short, we have studied batch size as an alternative temperature-like parameter, network architectures (DenseNet-40, VGG-19), datasets (CIFAR-100, SVHN), different pruning strategies (GLOBALMP), different optimizers (Adam) and different task (machine translation). In each case, we obtain results that are qualitatively similar to the results described in Section 3.2, thereby corroborating our main claim more generally.

4. Application of the Three-regime Model

In this section, we discuss several applications of our three-regime model.

4.1. Determining how to adjust temperature parameters

Here, we develop a scheme to tune temperature parameters based on the three-regime model. In particular, we are interested in the following question:

Problem statement. *Given a model trained with a particular temperature-like parameter and subsequently pruned to a target level of load, can we determine how to adjust the magnitude of temperature to improve performance?*

Table 1. The range of hyperparameters in the experiment setup: a dense model is trained with initial temperature T and subsequently pruned to target level of load L.

Initial temperature to adjust		Target level of load	
Training Epochs (T_{epoch})	Batch Size (T_{batch})	Density (L_{density})	Width Scalings (L_{width})
160	128	$L_{\text{density}} \subseteq \{5, 7, 10, 14, 20, 40\}\%$	$L_{\text{width}} \subseteq \{0.5, 1, 2, 4, 6, 8\}$

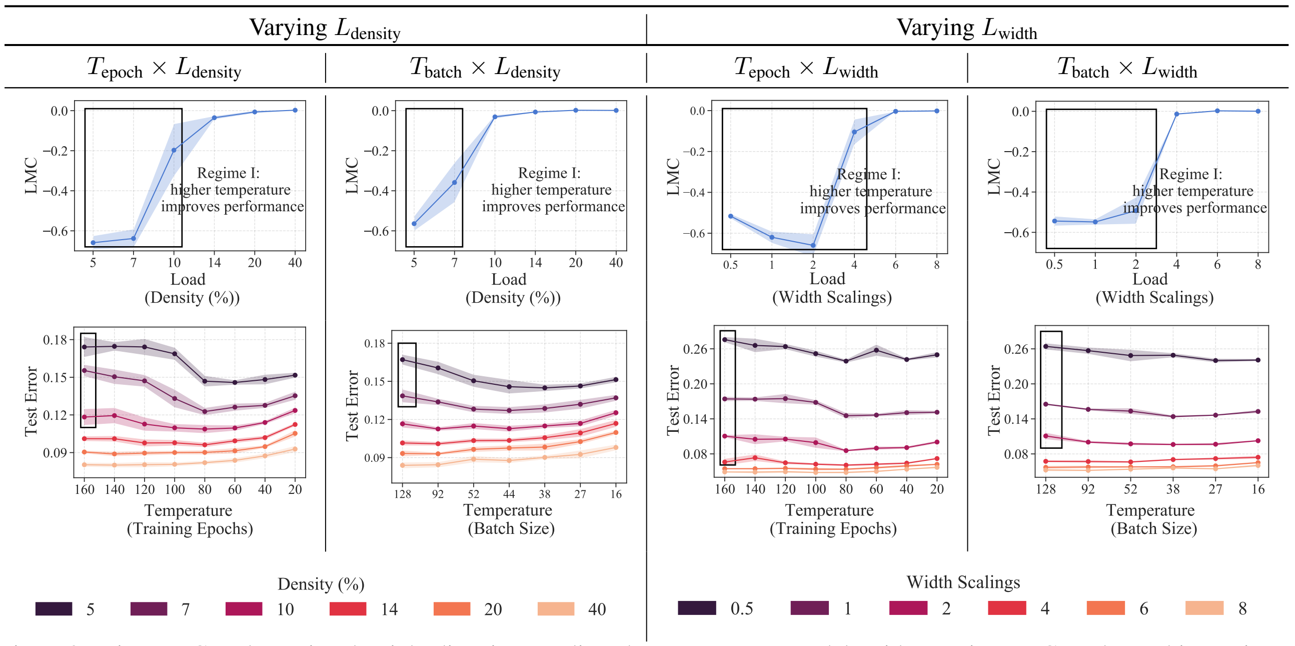


Figure 3. Using LMC to determine the right direction to adjust the temperature: models with negative LMC are located in Regime I (annotated by the black box), and their test error can be reduced by increasing temperature. Otherwise, models with close-to-zero LMC benefit from decreasing temperature. Note that a smaller training epoch or a smaller batch size corresponds to a higher temperature.

Experiment details. As a concrete setup, we explore two hyperparameters to represent the load-like parameter L of the pruned models: model density L_{density} and width scaling L_{width}. It is worth noting that the unstructured pruning method cannot alter the model width; thus, to vary L_{width} of the pruned model, we train multiple dense models with varying widths and prune them to identical densities.

Additionally, we consider two hyperparameters to represent the temperature-like parameter T to be adjusted in the first stage of dense model training: training epoch T_{epoch} and batch size T_{batch}. The ranges for these values are specified in Table 1. We use commonly used temperature values for our experiments: for CIFAR-10, a batch size of 128 and training epochs of 160 are commonly used. For instance, see Table 1 in Frankle et al. (2020); Liu et al. (2021). Thus, it is reasonable to assume that a practitioner might start with these initial temperature values similar to our study. We also note that when varying one specific parameter to control load, other control parameters are kept constant. For example, when varying L_{density}, L_{width} is fixed at 1, and while varying L_{width}, L_{density} is fixed at 5%. This is also applied when adjusting the temperature. The experiment is conducted using PreResNet-20 on the CIFAR-10 dataset, with the models being pruned using the UNIFORMMP.

Conventional wisdom. The conventional approach suggests that dense models should be trained to completion before being pruned (Lecun et al., 1989; Han et al., 2015).

Proposed method. We observe that the conventional wisdom of training to completion remains effective for pruned models belonging to Regime II, but loses its effectiveness for models in Regime I, where early stopping proves to be more useful. Note that recent studies also show improved pruning using early stopping (Li et al., 2020; Liu et al., 2021; Shen et al., 2022), while our method here gives a more comprehensive way to view this problem. Consequently, we propose a temperature-tuning method outlined in Algorithm 1. This algorithm determines whether training to completion (smaller temperature) or doing early stopping (larger temperature) is necessary to enhance the performance of the pruned model. Our approach assesses the LMC of the pruned models to identify whether they fall within the regime where conventional wisdom is less effective. If the current model belongs to Regime II (LMC $\in [\epsilon, 0]$), then

Algorithm 1 Temperature Tuning

```
Input: dense model initialization $\Theta_0$, initial temperature $T_0$ and increment $\Delta T$, target load $L$, training procedure $\mathrm{Train}$ (Section 2.2), pruning procedure $\mathrm{Prune}$ (Section 2.3), LMC threshold $\epsilon$
Output: Tuned temperature $T'$
Train with temperature: $\Theta$ = $\mathrm{Train}$ ($\Theta_0$, $T$)
Prune to target load: $\Theta \odot \mathcal{M}$ = $\mathrm{Prune}$ ($\Theta$, $L$)
Compute LMC on $\Theta \odot \mathcal{M}$ via Equation (1)
if LMC $< \epsilon$ then
    Regime I $= true$
    $T'$ = $T_0$ + $\Delta T$
else
    Regime I $= false$
    $T'$ = $T_0$
end if
```

Algorithm 2 Model Selection via LMC and test error

```
Input: a set of trained dense models $\{\Theta_i\}_{i=1}^n$, and their test errors $\{\mathrm{error}_{\text{test}}(\Theta_i)\}_{i=1}^n$, LMC threshold $\epsilon$, training procedure $\mathrm{Train}$ (Section 2.2), pruning procedure $\mathrm{Prune}$ (Section 2.3), target load $L$, retraining epochs $\alpha$
Output: dense model $\Theta_{i^*}$ to prune
$i^* = \arg\min_i \{\mathrm{error}_{\text{test}}(\Theta_i)\}_{i=1}^n$
$\Theta_{i^*} \odot \mathcal{M} = \mathrm{Prune}(\Theta_{i^*}, L)$
Compute LMC on $\Theta_{i^*} \odot \mathcal{M}$ via Equation (1)
if LMC $< \epsilon$ then
    $\Theta_i \odot \mathcal{M}_i = \mathrm{Train}(\mathrm{Prune}(\Theta_i, L), \alpha)$ for $i \in [1, n]$
    $i^* = \arg\min_i \{\mathrm{error}_{\text{test}}(\Theta_i \odot \mathcal{M}_i)\}_{i=1}^n$
end if
```

conventional wisdom is effective, and we do not increase the temperature. If the current model belongs to Regime I (LMC $< \epsilon$), conventional wisdom fails, and we increase the temperature. We select a threshold value $\epsilon = -0.05$, and discuss the selection of the threshold in Appendix C.1.

Results. The results are presented in Figure 3. In this figure, the label $T \times L$ signifies that the target model load for pruning is L, and the initial temperature setting is T (which is the temperature parameter that we need to adjust). The four columns represent four distinct cases arising from the use of two specific hyperparameters to characterize L and two other hyperparameters to characterize T, namely training epochs, batch size, density, and model width, as detailed in Table 1.

On the first row of Figure 3, we showcase the LMC results evaluated on a pruned model that has been trained with temperature T and pruned to load L. Our method (Algorithm 1) then identifies the regime to which the pruned model belongs and determines if increasing T is beneficial for different L values. On the second row, we present the test error resulting from increasing the temperature for all load values, which is used to verify whether Algorithm 1 selects the appropriate L for increasing T by examining if the test error decreases when T is increased.

From Figure 3, we see that Algorithm 1 effectively identifies that multiple pruning settings in Table 1 necessitate adjustments with higher temperatures. For instance, see the case $T_{\text{epoch}} \times L_{\text{density}}$. The LMC results in the first row indicate that three out of six L_{density}, specifically {5, 7, 10}%, have an LMC $< \epsilon$ (highlighted by the black box), signifying that these settings (pruned models) belong to Regime I and require tuning with higher temperatures. Indeed, the test error results in the second row demonstrate that increasing the temperature on these settings can reduce the test error. On the other hand, the remaining L_{density} {14, 20, 40}% benefit from decreasing temperature. We can find similar results on the other three columns, i.e., depending on the LMC value, one can determine whether to increase temperature or not. In Appendix C.2, we study the third load parameter, model depth L_{depth}, and we get consistent results.

By utilizing the loss landscape measure LMC, our method efficiently determines the correct regime for hyperparameter tuning. Thus, our method provides an efficient approach to predict the more efficient direction of tuning hyperparameters based on a single initial (T, L) pruning setting, thereby eliminating the need for costly grid searches. In contrast to the grid search, which typically necessitates at least two dense model training runs with different hyperparameters, our method accomplishes this with a single dense model training run. Consequently, our method exhibits twice the efficiency of the grid search approach. Furthermore, our method works on a wide range of load-like parameters, making it applicable to different target model sizes.

4.2. Selecting the best model without grid search

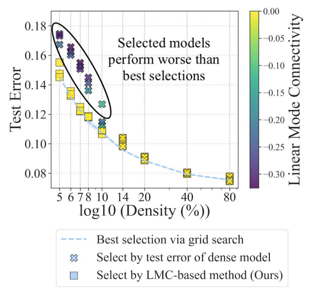
Here, we develop a model selection method based on the three-regime model. In particular, we are interested in the following question.

Problem statement. *Given a set of models trained with diverse magnitudes of temperature-like parameters (training epochs or batch sizes) and a target model density, which model should we prune to obtain optimal pruning performance?*

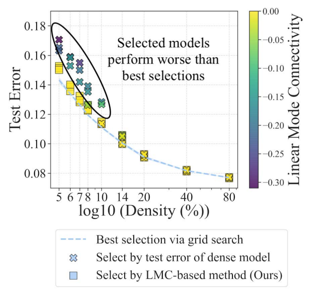
Experiment details. We study PreResNet-20 on CIFAR-10. We prune the model to 9 different model densities, as detailed in Section 3.1. For each target model density, we select a model to prune from two sets: 1) 16 different training epochs ranging from 10 to 160 and a fixed batch size; and 2) 13 different batch sizes ranging from 16 to 512 and a fixed training epoch.

Conventional wisdom. The conventional approach (Lecun et al., 1989; Han et al., 2015) to selecting a dense model to prune is based on the test error of the trained dense model.

Proposed method. We find that the conventional wisdom is effective when the pruned model belongs to Regime II while it does not work when it belongs to Regime I. Therefore, we propose an LMC-based model selection method (Algorithm 2), which uses the LMC metric to detect the wrong prediction made by conventional wisdom. Our approach uses the LMC metric to evaluate whether the dense model chosen by conventional means falls within the undesirable regime. If the model belongs to Regime II (LMC $\in [\epsilon, 0]$), conventional wisdom is effective, and the selection process concludes. Conversely, if the model falls under Regime I (LMC $< \epsilon$), conventional wisdom fails, and we evaluate alternative candidate models. We set LMC threshold $\epsilon = -0.05$, same as in Section 4.1. We set the retraining epochs $\alpha = 2$, which is much smaller than the 160 retraining epochs used by the grid search method. We also consider the realistic case that the evaluation has no access to test data and propose an LMC-and-CKA-based method Algorithm 2.1, which only requires access to training data.



(a) Selecting the best model from those trained with various training epochs.



(b) Selecting the best model from those trained with various batch sizes.

Figure 4. Selecting temperature using the LMC-based method (squares) leads to a smaller test error than selecting temperature using the test error of the unpruned dense model (crosses). The performance of LMC-based selection is close to the best test error found by grid search (dashed lines). (Left) Selecting the best training epoch. (Right) Selecting the best batch size. Models that perform significantly worse than grid search tend to have worse LMC, shown by the dark color of markers.

Results. We compare three model selection approaches, i.e., selecting by test error of dense model (conventional wisdom), selecting jointly by LMC and test error (ours, Algorithm 2), and best selections obtained by grid search that conducts the full train-prune-retrain procedure for all models in the set. The results of comparing the three model selection methods are presented in Figure 4. The x axis represents the model density, y axis represents the final test error of the pruned models after full time of retraining. The color of the markers represents the value of LMC measured on the pruned models. Figure 4a shows the results of selecting the best training epoch, while Figure 4b shows the results of selecting the best batch size.

In both figures, we observe the following. 1) The baseline method (select by test error of the dense model) demonstrates distinct performance in different regimes: it performs well in the large-density regime but poorly in the low-density regime, which is precisely characterized by the LMC measure (clear color transition from yellow to dark blue). 2) Our proposed LMC-based method achieves comparable performance in terms of final test error compared to grid search while significantly reducing the computational requirements. This efficiency is attributed to utilizing the LMC measure to determine whether the pruning configuration belongs to a regime where the baseline approach is effective. Specifically, in the high-density regime ($\{14, 20, 40, 80\}\%$), the LMC measure informs us to adopt the baseline solution without evaluating other candidates, thus avoiding unnecessary computations. Conversely, in the low-density regime ($\{5, 6, 7, 8\}\%$), we bypass the poor baseline solution identified by a low LMC value and perform computations (retraining 2 epochs for each candidate) to identify superior models with a high LMC value. The additional results of studying the test data-free case using the LMC-and-CKA-based method are provided in Appendix C.3.

4.3. Tuning the temperature of the SAM method

Here, we use the three-regime model to diagnose a dichotomous effect of training dense models with the optimizer SAM (Foret et al., 2021), and we propose a hyperparameter tuning scheme to mitigate the negative effect.

Problem statement. *Given a target model density, how can we tune the hyperparameter of SAM optimizer to train the dense model for optimal pruning performance?*

As explained in Section 2.1, SAM can be considered a high-temperature training method, with the neighborhood size (ρ) serving as the hyperparameter controlling the effective temperature. Previous research has demonstrated that incorporating SAM as an optimizer during the initial stage of training (Na et al., 2022) or utilizing SAM-inspired parameter perturbation-based optimizers (Peste et al., 2023) enhances pruning. In this study, we find that the SAM has a dichotomous effect when applied to pruning, i.e., SAM can either improve or damage pruning depending on the regime to which the pruned model belongs. This phenomenon can be explained by our proposed three-regime model. Subsequently, we demonstrate that the negative impact of SAM can be mitigated by adaptively tuning ρ using our three-regime model.

Experiment details. We study the effect of SAM on different regimes and vary the regimes of pruned models by

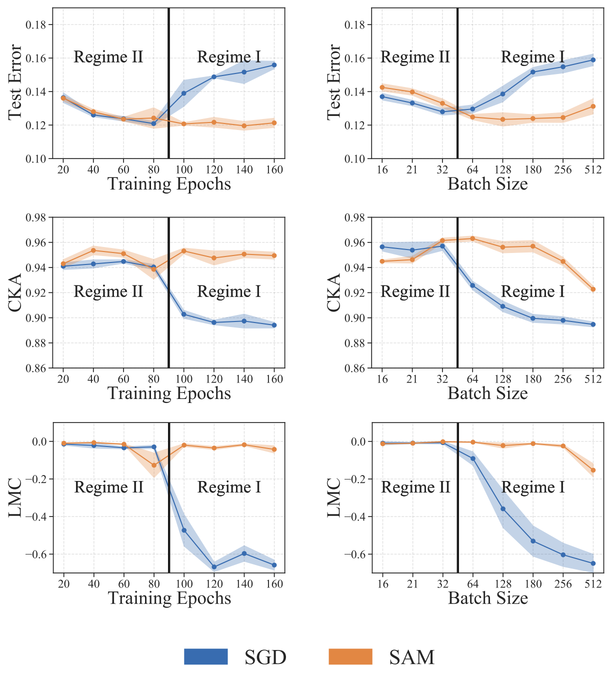


Figure 5. Comparing final pruning performance with dense model training using SGD versus SAM in Regime I and Regime II.

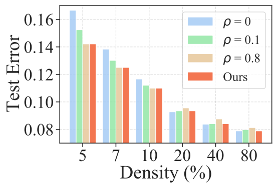
varying temperature-like or load-like parameters. In the first experiment, we vary the regimes between Regime I and II by changing the temperature, i.e., models are trained with a varying magnitude of temperature (batch size and training epochs) and pruned to a fixed density of 7% using UNIFORMMP. We set $\rho = 0.6$ for SAM. In the second experiment, we vary the regimes between Regime I and II by changing the load, i.e., models are pruned to varying model densities and trained with a fixed magnitude of temperature. We also show a simple CKA-based method that can adaptively tune the ρ of SAM for optimal pruning performance.

Conventional wisdom. Using SAM for dense model training is consistently beneficial for pruning performance (Na et al., 2022).

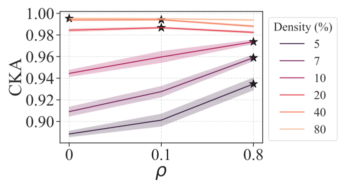
Proposed method. We find that the conventional wisdom is effective (SAM improves pruning) when the pruned model belongs to Regime I, while it damages pruning when the model belongs to Regime II and uses a large temperature (ρ), as this leads to an unfavorable transition from Regime II-B to a worse Regime II-A. We propose a CKA-based hyperparameter tuning method to adaptively tune the ρ of SAM according to the CKA similarity. Specifically, for a given model density, we assess the CKA similarity of the pruned models, and we choose the ρ that yields the highest CKA, thereby guiding the pruned models towards a more optimal Regime II-B.

Results. The results of the first experiment are shown in Figure 5. We show that when the SGD-based pruning settings (blue lines) fall within Regime I (poor LMC and CKA), SAM (orange lines) significantly reduces the test error, as well as improving LMC and CKA. This aligns with the conventional wisdom that SAM is effective in improving model compressibility, as well as our three-regime model that a large temperature is beneficial to pruning in Regime I. However, in Regime II (near-zero LMC and benign CKA), SAM yields negligible improvements, which means conventional wisdom loses its effectiveness in this regime.

The results of the second experiment are shown in Figure 6. From Figure 6a, we find that the low-density models benefit from using SAM with larger ρ, while high-density models suffer from using larger ρ: for densities 5%, 7%, and 10%, compared with $\rho = 0$ or $\rho = 0.1$, larger ρ (0.8) significantly reduces the test error, while for densities 20%, 40%, and 80%, larger ρ makes the test error a bit worse. Figure 6b demonstrates our CKA-based tuning method of selecting the ρ under different model densities based on the CKA values, and the red bar in Figure 6a shows that our approach always achieves optimal test error in tuning the ρ.



(a)



(b)

Figure 6. (a) Comparing the test error of small/large ρ hyperparameter and proposed CKA-based hyperparameter tuning method. (b) The proposed hyperparameter tuning method selects the ρ with the highest CKA value annotated by the black star markers.

5. Conclusion

Based on recent work in the statistical mechanics of learning, we have proposed a three-regime model of network pruning, which depends on temperature-like and load-like parameters. We discover a dichotomous phenomenon that arises when temperature-like and load-like parameters are varied: a higher temperature (used during the dense model training stage) results in good pruned model performance in heavily pruned networks, while a lower temperature (used during the dense model training stage) hurts final performance for lightly pruned networks. We find that popular metrics (such as the LMC and CKA similarity) closely track the transitions between regimes, providing operational ways of improving pruning. We anticipate the generalizability of our results and the reproducibility of the multi-regime taxonomy of load and temperature on other model compression techniques such as quantization and distillation.

Acknowledgments. We would like to acknowledge the DOE, IARPA, NSF, and ONR for providing partial support of this work.

References

Bahri, Y., Kadmon, J., Pennington, J., Schoenholz, S., Sohl-Dickstein, J., and Ganguli, S. Statistical mechanics of deep learning. *Annual Review of Condensed Matter Physics*, 11:501–528, 2020.

Baity-Jesi, M., Sagun, L., Geiger, M., Spigler, S., Arous, G. B., Cammarota, C., LeCun, Y., Wyart, M., and Biroli, G. Comparing dynamics: Deep neural networks versus glassy systems. In *International Conference on Machine Learning*, 2018.

Ballard, A. J., Das, R., Martiniani, S., Mehta, D., Sagun, L., Stevenson, J. D., and Wales, D. J. Energy landscapes for machine learning. *Physical Chemistry Chemical Physics*, 19(20):12585–12603, 2017.

Barbier, J., Krzakala, F., Macris, N., Miolane, L., and Zdeborová, L. Optimal errors and phase transitions in high-dimensional generalized linear models. *Proceedings of the National Academy of Sciences*, 116(12):5451–5460, 2019.

Barra, A. and Guerra, F. About the ergodic regime in the analogical Hopfield neural networks: moments of the partition function. *Journal of mathematical physics*, 49 (12):125217, 2008.

Barra, A., Bernacchia, A., Santucci, E., and Contucci, P. On the equivalence of Hopfield networks and Boltzmann machines. *Neural Networks*, 34:1–9, 2012.

Barsbey, M., Sefidgaran, M., Erdogdu, M. A., Richard, G., and Simsekli, U. Heavy tails in sgd and compressibility of overparametrized neural networks. In *Advances in Neural Information Processing Systems*, 2021.

Blalock, D., Gonzalez Ortiz, J. J., Frankle, J., and Guttag, J. What is the state of neural network pruning? In *Proceedings of Machine Learning and Systems*, 2020.

Bojar, O., Buck, C., Federmann, C., Haddow, B., Koehn, P., Leveling, J., Monz, C., Pecina, P., Post, M., Saint-Amand, H., Soricut, R., Specia, L., and Tamchyna, A. Findings of the 2014 workshop on statistical machine translation. In *Proceedings of the Ninth Workshop on Statistical Machine Translation*, 2014.

Brooks, C., Onuchic, J., and Wales, D. Statistical thermodynamics - taking a walk on a landscape. *Science (New York, N.Y.)*, 293:612–3, 08 2001.

Brush, S. G. History of the Lenz-Ising model. *Reviews of Modern Physics*, 39:883–893, 1967.

Draxler, F., Veschgini, K., Salmhofer, M., and Hamprecht, F. Essentially no barriers in neural network energy landscape. In *International Conference on Machine Learning*, 2018.

Engel, A. and den Broeck, C. P. L. V. *Statistical mechanics of learning*. Cambridge University Press, New York, NY, USA, 2001.

Evci, U., Pedregosa, F., Gomez, A., and Elsen, E. The difficulty of training sparse neural networks. In *International Conference on Machine Learning 2019 Workshop on Identifying and Understanding Deep Learning Phenomena*, 2019.

Foret, P., Kleiner, A., Mobahi, H., and Neyshabur, B. Sharpness-aware minimization for efficiently improving generalization. In *International Conference on Learning Representations*, 2021.

Frankle, J., Dziugaite, G. K., Roy, D., and Carbin, M. Linear mode connectivity and the lottery ticket hypothesis. In *International Conference on Machine Learning*, 2020.

Garipov, T., Izmailov, P., Podoprikhin, D., Vetrov, D. P., and Wilson, A. G. Loss surfaces, mode connectivity, and fast ensembling of DNNs. In *Advances in Neural Information Processing Systems*, 2018.

Han, S., Pool, J., Tran, J., and Dally, W. Learning both weights and connections for efficient neural network. In *Advances in Neural Information Processing Systems*, 2015.

Haussler, D., Kearns, M., Seung, H. S., and Tishby, N. Rigorous learning curve bounds from statistical mechanics. *Machine Learning*, 25(2):195–236, 1996.

He, K., Zhang, X., Ren, S., and Sun, J. Identity mappings in deep residual networks. In *European Conference on Computer Vision*, 2016.

Hopfield, J. Neural networks and physical systems with emergent collective computational abilities. *Proceedings of the National Academy of Sciences of the United States of America*, 79:2554–8, 05 1982.

Huang, G., Liu, Z., Van Der Maaten, L., and Weinberger, K. Q. Densely connected convolutional networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2017.

Kornblith, S., Norouzi, M., Lee, H., and Hinton, G. Similarity of neural network representations revisited. In *International Conference on Machine Learning*, 2019.

Krizhevsky, A., Nair, V., and Hinton, G. Cifar-10 and cifar-100 datasets. 2009.

Le, D. H. and Hua, B.-S. Network pruning that matters: A case study on retraining variants. In *International Conference on Learning Representations*, 2021.

Lecun, Y., Denker, J., and Solla, S. Optimal brain damage. In *Advances in Neural Information Processing Systems*, 1989.

Lee, J., Park, S., Mo, S., Ahn, S., and Shin, J. Layer-adaptive sparsity for the magnitude-based pruning. In *International Conference on Learning Representations*, 2021.

Li, Z., Wallace, E., Shen, S., Lin, K., Keutzer, K., Klein, D., and Gonzalez, J. Train big, then compress: Rethinking model size for efficient training and inference of transformers. In *International Conference on Machine Learning*, 2020.

Liu, S., Chen, T., Chen, X., Atashgahi, Z., Yin, L., Kou, H., Shen, L., Pechenizkiy, M., Wang, Z., and Mocanu, D. C. Sparse training via boosting pruning plasticity with neuroregeneration. In *Advances in Neural Information Processing Systems*, 2021.

Liu, Z., Sun, M., Zhou, T., Huang, G., and Darrell, T. Rethinking the value of network pruning. In *International Conference on Learning Representations*, 2019.

Martin, C. H. and Mahoney, M. W. Rethinking generalization requires revisiting old ideas: statistical mechanics approaches and complex learning behavior. *arXiv preprint arXiv:1710.09553*, 2017.

Martin, C. H. and Mahoney, M. W. Post-mortem on a deep learning contest: a Simpson's paradox and the complementary roles of scale metrics versus shape metrics. Technical Report Preprint: arXiv:2106.00734, 2021a.

Martin, C. H. and Mahoney, M. W. Implicit self-regularization in deep neural networks: Evidence from random matrix theory and implications for learning. *Journal of Machine Learning Research*, 22(165):1–73, 2021b.

Martin, C. H., Peng, T. S., and Mahoney, M. W. Predicting trends in the quality of state-of-the-art neural networks without access to training or testing data. *Nature Communications*, 2021.

McCandlish, S., Kaplan, J., Amodei, D., and Team, O. D. An empirical model of large-batch training. *arXiv preprint arXiv:1812.06162*, 2018.

Molchanov, P., Tyree, S., Karras, T., Aila, T., and Kautz, J. Pruning convolutional neural networks for resource efficient inference. In *International Conference on Learning Representations*, 2017.

Na, C., Mehta, S. V., and Strubell, E. Train Flat, Then Compress: Sharpness-Aware Minimization Learns More Compressible Models. In *Findings of the Conference on Empirical Methods in Natural Language Processing*, 2022.

Peste, A., Vladu, A., Kurtic, E., Lampert, C., and Alistarh, D. Cram: A compression-aware minimizer. *International Conference on Learning Representations (ICLR)*, 2023.

Renda, A., Frankle, J., and Carbin, M. Comparing rewinding and fine-tuning in neural network pruning. In *International Conference on Learning Representations*, 2020.

Rosenfeld, J. S., Frankle, J., Carbin, M., and Shavit, N. On the predictability of pruning across scales. In *International Conference on Machine Learning*, 2021.

Sermanet, P., Kavukcuoglu, K., and LeCun, Y. Traffic signs and pedestrians vision with multi-scale convolutional networks. In *Snowbird Machine Learning Workshop*, 2011.

Seung, H. S., Sompolinsky, H., and Tishby, N. Statistical mechanics of learning from examples. *Physical Review A*, 45(8):6056–6091, 1992.

Shen, M., Molchanov, P., Yin, H., and Alvarez, J. M. When to prune? a policy towards early structural pruning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022.

Simonyan, K. and Zisserman, A. Very deep convolutional networks for large-scale image recognition. In *International Conference on Learning Representations*, 2014.

Sorscher, B., Geirhos, R., Shekhar, S., Ganguli, S., and Morcos, A. Beyond neural scaling laws: beating power law scaling via data pruning. In *Advances in Neural Information Processing Systems*, 2022.

Stillinger, F. *Energy Landscapes, Inherent Structures, and Condensed-Matter Phenomena*. Princeton University Press, 2015.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., and Polosukhin, I. Attention is all you need. In *Advances in Neural Information Processing Systems*, 2017.

Wales, D. *Energy Landscapes: Applications to Clusters, Biomolecules and Glasses*. Cambridge Molecular Science. Cambridge University Press, 2003.

Watkin, T. L. H., Rau, A., and Biehl, M. The statistical mechanics of learning a rule. *Rev. Mod. Phys.*, 65(2): 499–556, 1993.

Yang, Y., Hodgkinson, L., Theisen, R., Zou, J., Gonzalez, J. E., Ramchandran, K., and Mahoney, M. W. Taxonomizing local versus global structure in neural network loss landscapes. In *Advances in Neural Information Processing Systems*, 2021.

Yang, Y., Theisen, R., Hodgkinson, L., Gonzalez, J. E., Ramchandran, K., Martin, C. H., and Mahoney, M. W. Evaluating natural language processing models with generalization metrics that do not need access to any training or testing data. *arXiv preprint arXiv:2202.02842*, 2022.

Yao, Z., Gholami, A., Keutzer, K., and Mahoney, M. W. PyHessian: Neural networks through the lens of the Hessian. In *IEEE International Conference on Big Data*, 2020.

A. Related Work

A.1. Overview of related work

Statistical mechanics of learning. Earlier work from the statistical mechanics of learning makes explicit connections between NN control parameters and load/temperature parameters, e.g., the temperature in the Boltzmann distribution (Seung et al., 1992; Watkin et al., 1993; Haussler et al., 1996; Engel & den Broeck, 2001; Bahri et al., 2020); and more recent work has argued that these should empirically affect the output of training (Martin & Mahoney, 2017). Based on this, Martin & Mahoney (2017) argues for adjusting load and temperature in mitigating overfitting, encouraging the model to exhibit Heavy-Tailed Self-Regularization (HT-SR), as its layer weight matrices develop stronger correlations over the course of training (Martin & Mahoney, 2021b). The study of the correlation among HT-SR-related generalization metrics, model, task and test performance has been used to identify Simpsons paradoxes in deep learning contests (Martin & Mahoney, 2021a), which finds positive correlation between metrics when tasks are coupled and negative on certain task when performed alone. Importantly, our more phenomenological approach statistical mechanics view of NN learning is different from the commonly used (classical or more recent) works (Seung et al., 1992; Watkin et al., 1993; Haussler et al., 1996). However, it is consistent with the theoretical foundations of Martin & Mahoney (2017; 2021b). Our results provide additional evidence, complementing that of Martin et al. (2021); Yang et al. (2022), that this more phenomenological approach to the statistical mechanics of learning can provide more principled basis for a practical theory of NN performance. Also, recently we see a surge of interest in using statistical mechanics to analyze and improve learning, including Baity-Jesi et al. (2018) on the glassy behavior of neural networks, Barbier et al. (2019) on optimal generalization error of generalized linear systems, and Sorscher et al. (2022) on selecting easy versus hard samples used in training.

Load and temperature. The notation of load and temperature was introduced earlier in the statistical mechanics of learning, e.g., the load in the Hopfield model of associative memory (Hopfield, 1982; Barra & Guerra, 2008; Barra et al., 2012) and the temperature in the Boltzmann distribution (Seung et al., 1992) and lsing model (Brush, 1967). As the load and temperature change, the learning system can change its performance and properties dramatically and qualitatively. Recent work studies the load and temperature in modern deep NN training. Several works characterize the temperature as the noise scale (or stochasticity) in the optimization process, specified as concrete hyperparameters that influence the training, e.g., the early stopping epochs (Martin & Mahoney, 2017), batch sizes, and learning rate (McCandlish et al., 2018; Yang et al., 2021). Martin & Mahoney (2017) characterizes the load-like parameter as the effective capacity of the model or the ratio of the number of data points to a parameter characterizing the complexity of the model. This line of work also shows that varying load and temperature can make the NN training display phase behavior. There is no prior work that explicitly and systematically studies the load and temperature in the task of NN pruning and further observes and uses the phase transition behavior. A few works have studied the effect of temperature-like parameters such as training epochs on network pruning. Li et al. (2020) finds that larger models trained short of convergence outperform small models trained to convergence. Although Li et al. (2020) gives a valuable heuristic on early stopping, the two hyperparameters training epochs and model size may be intertwined, and therefore, it is unclear whether early stopping is actually needed to reach the optimal pruning results. Shen et al. (2022) proposes a metric to indicate an early point to end the dense model training and begin pruning-retraining, showing that the model pruned at early epochs outperforms the one drawn at the later epoch. Liu et al. (2021) shows that when pruning happens during the early training phase with large learning rates, models can easily recover performance via retraining. These studies provide evidence that varying temperature-like parameters in dense model training have a significant effect on pruning performance. Via the VSDL model, our work provides a more explicit framework for these tasks.

Loss landscape. The study of loss landscapes has received attention among the machine learning community in recent years. Many of the important ideas have grounded in statistical mechanics and chemical physics (Brooks et al., 2001; Wales, 2003; Stillinger, 2015; Ballard et al., 2017). Recent work uses the loss landscape to analyze many modern techniques such as large-batch training (Yao et al., 2020) and pruning (Frankle et al., 2020; Evci et al., 2019). Yang et al. (2021) gives a systematic study on the local and global geometry properties of loss landscape in “load-temperature” framework. Their empirical finding shows that the global loss landscape metrics such as mode connectivity (Garipov et al., 2018) and model similarity (Kornblith et al., 2019) perform well in indicating the phase transition of model training, and the transition is closely correlated with the generalization performance of trained models. Frankle et al. (2020) uses linear mode connectivity to indicate the phase transition of drawing “lottery tickets” during dense model training. Both works show that the global property of loss landscape is quite effective in explaining different phase transitions observed.

A.2. Discussion on the difference between this work and Yang et al. (2021)

We discuss several aspects of our approach that differ from how Yang et al. (2021) applied the VSDL model.

First, the astute reader will have noticed that, since the pruning pipeline is

[Train-dense-model – Prune – Retrain-pruned-model],

there actually exists four variables of loads and temperatures that could potentially influence the pruning performance: 1) load of dense-model; 2) temperature of dense-model; 3) load of pruned-model; and 4) temperature of pruned-model. Our main empirical finding is that, when one targets a given load of the pruned model, one should adjust the temperature of the dense model based on the phase behavior of the global loss landscape of pruned models. That is, in this special multiple-stage pipeline, where a large perturbation (pruning) is involved in the middle stage, changing the temperature in the first stage will significantly influence (and can be used to control) the global structure in the final stage. Our modeling results demonstrate that the "load-temperature" framework of the VSDL model can simplify and help solve a complex multi-stage problem that is commonly seen in applications. Other possible examples of this include transfer learning, distillation, quantization, etc.

Second, the astute reader may also wonder what is the role of Hessian information, given its importance in how Yang et al. (2021) taxonomized loss landscapes. In Yang et al. (2021), Hessian information was used to identify the sharp transition at the interpolation threshold. In our study of pruning, however, most of the pruned models are unable to interpolate the training data. Compared with Figure 3 in Yang et al. (2021), we observed empirically that the Hessian eigenvalues are too large to reach locally flat phases. That is, we found that using metrics from Yang et al. (2021) that were used to measure the *local structure* of loss landscapes (such as Hessian trace or leading Hessian eigenvalues) is not as informative as the global LMC and CKA metrics in the pruning setting, and it does not significantly help identify the different regimes for the pruning problem.

Finally, we would like to emphasize the novelties of our work in comparison to Yang et al. (2021): 1) While Yang et al. (2021) builds a theory-driven framework, our paper proposes practical machine learning algorithms, including hyperparameter tuning and model selection, to address the specific challenges in pruning. 2) While Yang et al. (2021) primarily focuses on taxonomizing various types of phase transitions, we leverage these phase transitions to design effective pruning algorithms with direct implications for practical applications.

B. Additional Details for Three-regime Taxonomy

B.1. Implementation details

Datasets. For the image classification task, we consider CIFAR-10, CIFAR-100 (Krizhevsky et al., 2009), and SVHN (Sermanet et al., 2011). CIFAR-10 comprises 50,000 training images and 10,000 testing images with 10 categories. CIFAR-100 comprises 50,000 training images and 10,000 testing images with 100 categories. SVHN comprises 73,257 training images and 26,032 testing images with 10 categories. For the machine translation task, we consider WMT14 (Bojar et al., 2014) German to English (DE-EN) dataset. We subsample 1.28M sentence pairs from the WMT14 training set for training and report the BLEU score on the validation set.

Model architectures. For the image classification task, we use PreResNet-20 (Pre-Activation ResNet, (He et al., 2016)), DenseNet-40 (Huang et al., 2017), and VGG-19 (Simonyan & Zisserman, 2014). Our implementation is based on Liu et al. (2019). For the machine transition task, we use Transformer-base (Vaswani et al., 2017).

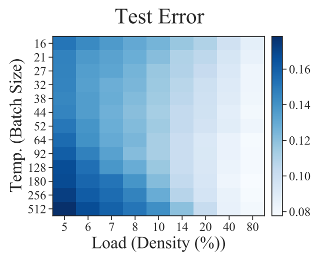
Training procedures. For full model training and pruned model retraining, we use SGD as the default optimizer. The default hyperparameters include a momentum of 0.9, weight decay of 1e-4, and a training duration of 160 epochs. Learning rate decay is applied with an initial learning rate of 0.1, which decreases by a factor of 10 at epochs 80 and 120. In the study of varying training epochs as temperature, we fix batch size as 64. In the study of varying batch sizes as temperature, we fix the training iterations as 62400. For each configuration, we train the dense model with three different random seeds and retrain the pruned sparse model with three random seeds as well. We report the average test error and loss landscape measures across these random seeds.

Pruning procedures. We implement the unstructured magnitude pruning methods, UNIFORMMP and GLOBALMP, following Liu et al. (2019); Lee et al. (2021). For image classification models, we only prune weights in convolution layers, as suggested by Liu et al. (2019), while keeping the last linear layer intact due to its relatively small number of parameters.

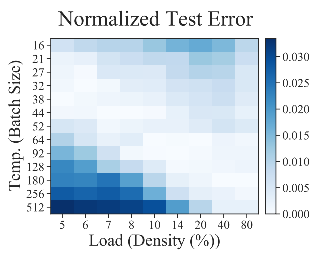
Hyperparameters for different metrics. We implement the LMC and CKA similarity metrics based on Yang et al. (2021). For LMC measurement, we use the entire training set, while for CKA similarity measurement, we draw 6,400 samples from the training set.

B.2. Additional Supporting Results for Three-regime Taxonomy

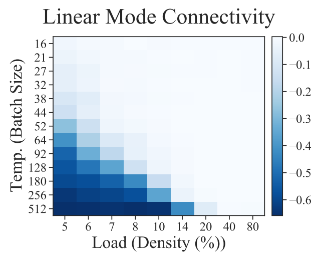
Different temperature parameters. We corroborate our main claim by studying batch size as an alternative temperature parameter, in addition to adjusting the number of training epochs. The results are shown in Figure 7. Comparing Figure 7 with Figure 2, we see that the three regimes are still present and the observation that high temperature improves pruning for poorly-connected Regime I while hurts pruning for well-connected Regime II-B also holds. Therefore, our central claim remains consistent even with the introduction of different temperature-like parameters.



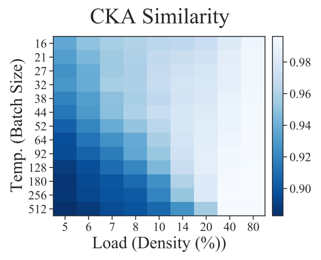
(a) Test error



(b) Normalized test error



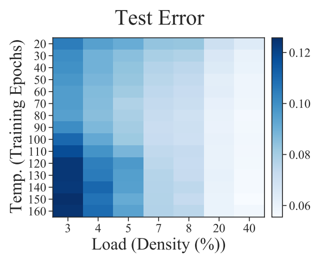
(c) LMC



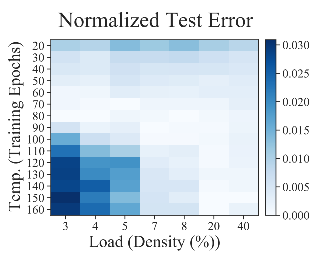
(d) CKA similarity

Figure 7. **(Batch size as temperature).** Partitioning the 2D diagram of model density – batch size into three regimes. Models are trained with PreResNet-20 on CIFAR-10.

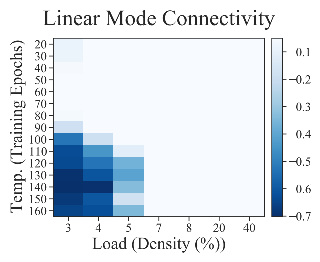
Different architectures. We extend the experiments with DenseNet-40 and VGG-19 on CIFAR-10 in Figure 8 and Figure 9. The experiment produces analogous results to Figure 2 and supports our main claim. We note that DenseNet-40 and VGG-19 are larger architectures with more parameters (1M and 20M parameters) than the basic architecture PreResNet-20 (0.27M parameters), so we are able to prune it to a smaller density.



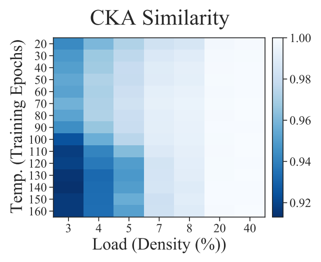
(a) Test Error



(b) Normalized test error

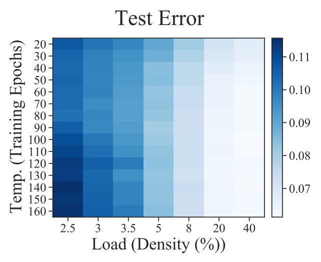


(c) LMC

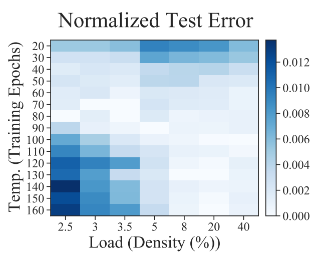


(d) CKA similarity

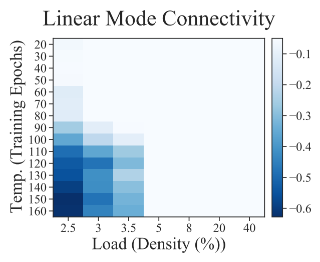
Figure 8. **(DenseNet-40).** Partitioning the 2D model density – training epochs diagram into three regimes. Models are trained with DenseNet-40 on CIFAR-10.



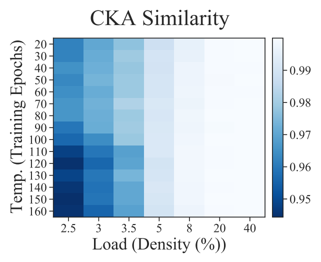
(a) Test error



(b) Normalized test error



(c) LMC



(d) CKA similarity

Figure 9. **(VGG-19).** Partitioning the 2D model density – training epochs diagram into three regimes. Models are trained with VGG-19 on CIFAR-10.

Different datasets. We extend the experiments with different datasets CIFAR-100 and SVHN in Figure 10 and Figure 11. The experiment produces consistent results with Figure 2 and supports our main claim.

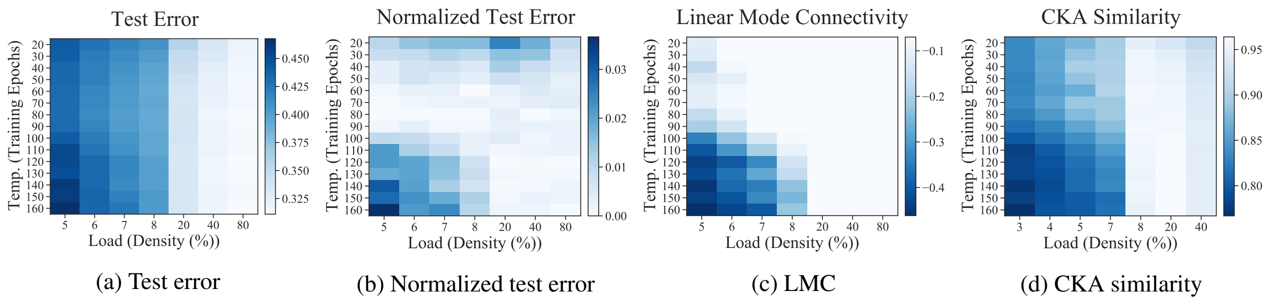


Figure 10. **(CIFAR-100).** Partitioning the 2D diagram of model density—training epochs into three regimes. Models are trained with PreResNet-20 on CIFAR-100.

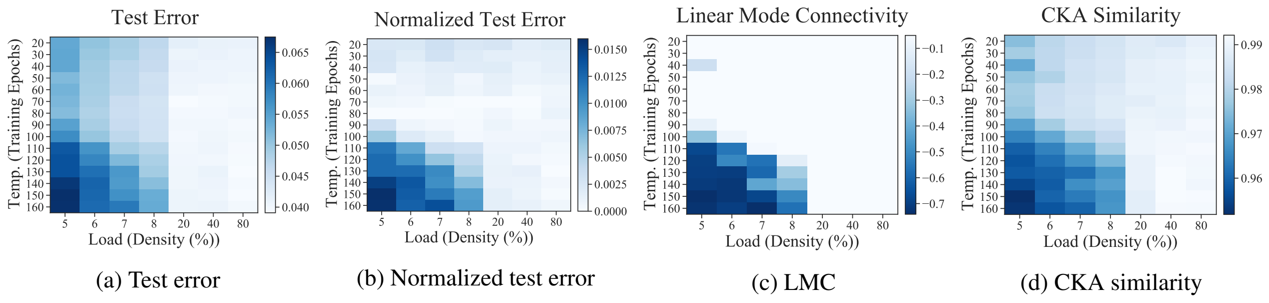


Figure 11. **(SVHN).** Partitioning the 2D diagram of model density—training epochs into three regimes. Models are trained with PreResNet-20 on SVHN.

Different pruning strategy. We confirm the three-regime taxonomy still holds under a different pruning strategy, namely performing GLOBALMP in place of UNIFORMMP, as shown in Figure 12. We maintain a similar experimental setup, varying training epochs to adjust the temperature, and we produce the same set of 2D diagrams as Figure 2. Comparing Figure 12 with Figure 2, we see that the three regimes are visibly present, and the higher temperature has a dichotomous effect on test error depending on whether LMC is negative or not. Therefore, our central claim is robust to the choice of the pruning method.

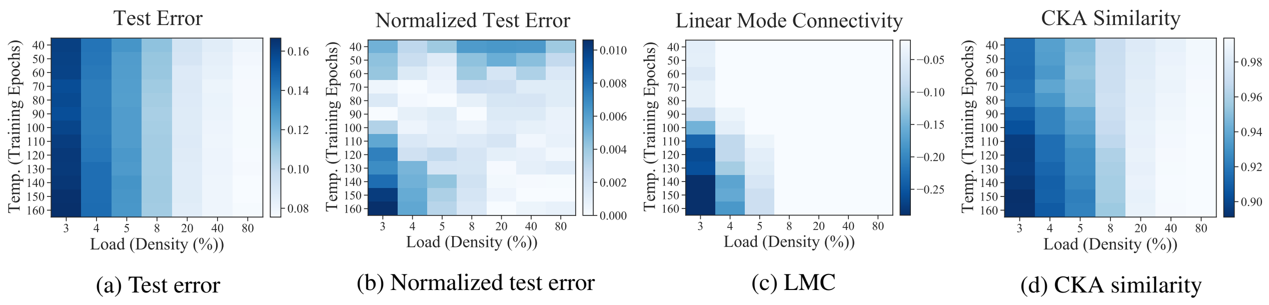
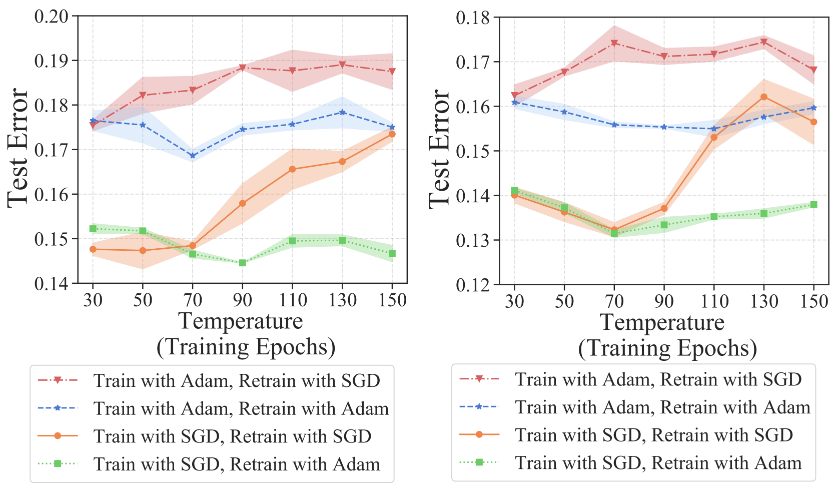


Figure 12. **(Global magnitude pruning).** Partitioning the 2D model density—training epochs diagram into three regimes where we use global magnitude pruning (GLOBALMP) instead of uniform magnitude pruning (UNIFORMMP). Models are trained with PreResNet-20 on CIFAR-10.

Different optimizer. We conducted ablation studies on SGD versus Adam with four different settings for network pruning: 1) dense model training with Adam, retrain with SGD, 2) dense model training with Adam, retrain with Adam, 3) dense model training with SGD, retrain with SGD, 4) dense model training with SGD, retrain with Adam. All four settings focus on heavily-pruned models (pruned to the density of 5%, 6%), and we expect to see that a large temperature potentially helps improve the test error.

In more detail, here is the experimental setup. We trained PreResNet-20 on CIFAR-10, using the Adam optimizer with 0.001 as the initial learning rate and the SGD optimizer with 0.1 as the initial learning rate. The weight decay is 1e-4 for both. The setup of the learning rate decay schedule, model architecture, dataset, and pruning is the same as the setup in the main paper. We grid-searched the initial learning rate of Adam using values from 0.1, 0.01, 0.001, 0.0001. We trained the dense model with one random seed and fine-tuned each pruned model with three random seeds.

The results are presented in Figure 13. First, we noticed that reducing the training epochs (using a large temperature) improves the test error for all four settings, which is consistent with our main claim. Additionally, we noticed that Adam-based dense model training does not work as well as SGD, which is observed in the literature. Interestingly, retraining with Adam significantly improves our result (see the green curves). Our interpretation is that Adam during the retraining works by stabilizing the training with heavily-pruned bottleneck layers.



(a) Density = 5% (b) Density = 6%

Figure 13. **(Adam optimizer for pruning.)** Using a large temperature (reducing the training epochs) improves the test error of heavily pruned models consistently when using Adam and/or SGD optimizers during pruning. Models are trained with PreResNet-20 on CIFAR-10.

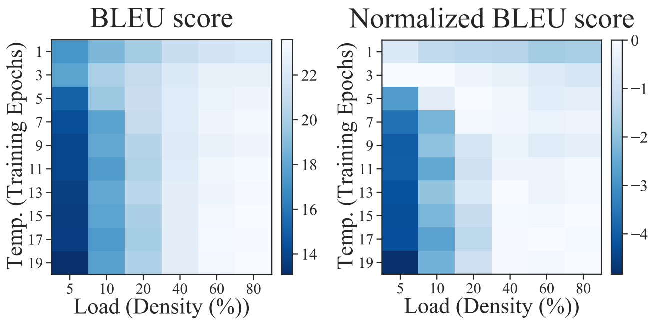
Different task. We show that our model extends to machine translation by performing an experiment on the WMT14 (Bojar et al., 2014) German to English (DE-EN) dataset using the standard Transformer (Vaswani et al., 2017) model. We use the Adam optimizer with an inverse square-root learning rate schedule and 4000 warm-up iterations for training and fine-tuning the model. We subsample 1.28M sentence pairs from the WMT14 training set and report the validation BLEU score. The Transformer-base model has 6 layers, 8 attention heads, and an embedding dimension of 512. We varied the temperature over 10 training epochs and the load over 6 densities, ranging from 5% to 80%, using UNIFORMMP defined in the paper. We trained the dense model with one random seed and retrain each pruned model with three random seeds.

The results are presented in Figure 14. They demonstrate the same dichotomous phenomenon that the large temperature enhances the BLEU score in low densities, while smaller temperatures help high-density settings, which is consistent with our main findings in image classification tasks.

C. Additional Details on Three-regime based Applications

C.1. Determining the LMC threshold value

In Section 4.1 and Section 4.2, we consistently use -0.05 as the LMC threshold value ϵ, which is used to distinguish whether a pruned model belongs to Regime I or Regime II-B (see Figure 1b) in our proposed Algorithm 1 and Algorithm 2. This threshold is empirically determined by our three-regime taxonomy results of different datasets and architectures in Section 3.2. Among the three-regime results of multiple architectures and datasets, we consistently find that the transition between Regime I and II-B is very sharp: Regime I, which is the darker region on the lower left, always has a much smaller LMC than Regime II-B, the LMC changing quickly from close to zero to lower than -0.4 when the transition happens. Therefore, -0.05 is a reasonable critical value and is sufficient to distinguish the two regimes. Furthermore, our proposed



(a) BLEU score (b) Normalized BLEU score

Figure 14. **(Machine transition.)** Larger temperature (fewer training epochs) enhances the BLEU score in low densities, while smaller temperatures help high-density settings. Models are trained with Transformer-base on WMT14.

methods are robust to the choices of the threshold. Researchers who apply our algorithms in Section 4.1 and Section 4.2 also do not need to generate the three-regime taxonomy results again to determine the threshold value. In other words, they can apply our prior observation in Section 3.2 and employ the threshold of -0.05 as a hyperparameter initialization for their specific tasks.

Table 2. The range of hyperparameters considered in the experiment setup: a dense model is trained with initial temperature T and subsequently pruned to target level of load L.

Initial temperature to adjust		Target level of load
Training Epochs (T_{epoch})	Batch Size (T_{batch})	Model Depth (L_{depth})
160	128	$L_{\text{depth}} \subseteq \{20, 32, 44, 56, 80, 152\}$

C.2. Additional results on hyperparameter-tuning scheme

This subsection provides supplementary results to Section 4.1, demonstrating how to use the proposed temperature-tuning scheme to predict if we should increase or decrease temperature given a specific pruning configuration with load and temperature. We study model depth T_{depth} as a load-like parameter.

Experiment details. We study PreResNet-20 with width scaling 1 on CIFAR-10 and prune the model to 5% of model density with UNIFORMMP. We consider studying the varying temperature T and load L, as detailed in Table 2.

Results. Results are presented in Figure 15. From the LMC results on the first row, our scheme determines that all the depths should be tuned with higher temperatures since they have LMC $< \epsilon$ (ϵ = -0.05, annotated by the black box). This prediction is verified by the test error results on the second row, which shows that the test error can be reduced by increasing the temperature. In contrast with the other two load-like parameters density and width scaling, one special finding here is that increasing model depth does not help the configuration transit to Regime II (and thus all the cases studied can be benefited from higher temperatures).

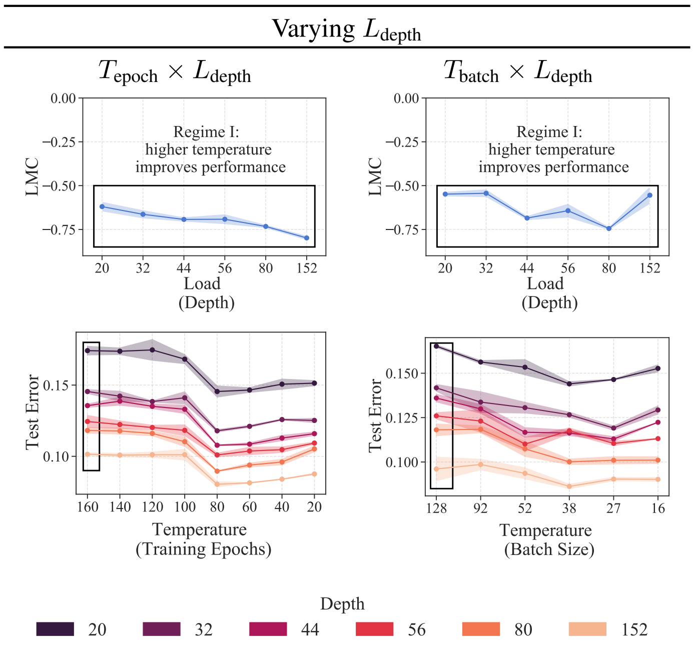


Figure 15. Using LMC to determine whether to increase or decrease temperature: models with negative LMC are located in Regime I (annotated by the black box), and their test error can be reduced by increasing temperature.

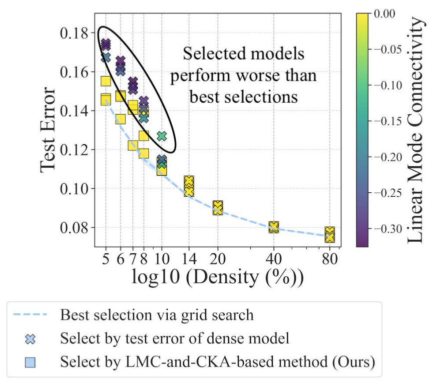
Algorithm 2.1 Model Selection via LMC and CKA

```
Input: a set of trained dense models $\{\Theta_i\}_{i=1}^n$, and corresponding test errors $\{\text{error}_{\text{test}}(\Theta_i)\}_{i=1}^n$, LMC threshold $\epsilon$, training procedure Train (Section 2.2), pruning procedure Prune (Section 2.3), target load $L$, retraining epochs $\alpha$
Output: dense model $\Theta_{i^*}$ to prune
$i^* = \arg\min_i\{\text{error}_{\text{test}}(\Theta_i)\}_{i=1}^n$
$\Theta_{i^*} \odot \mathcal{M} = \text{Prune}(\Theta_{i^*}, L)$
Compute LMC on $\Theta_{i^*} \odot \mathcal{M}$ via Equation (1)
if LMC $< \epsilon$ then
    $\Theta_i \odot \mathcal{M}_i = \text{Train}(\text{Prune}(\Theta_i, L), \alpha)$ for $i \in [1, n]$
    $i^* = \arg\max_i\{\text{CKA}(\Theta_i \odot \mathcal{M}_i)\}_{i=1}^n$ via Equation (2)
end if
```

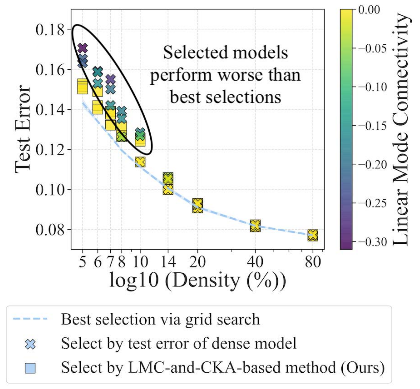
C.3. Additional details for model selections

In this subsection, we provide diagrams to describe the algorithms used in model selection tasks. Algorithm 2 shows the proposed method of using LMC and test error for selection with access to test data. Algorithm 2.1 shows the proposed method using LMC and CKA for selection without access to test data.

We provide the results of evaluating Algorithm 2.1. See Figure 16. In the high-density regime, this method shows comparable performance as the grid search but uses fewer retraining epochs: this method takes 2 retraining epochs to determine that the baseline selection performs well for this density, while the grid search takes 160 retraining epochs for each candidate in the set. In the low-density regime, this method performs better than the baseline selections by using the LMC to avoid selecting the bad model to prune.



(a) Selecting the best model from those trained with various training epochs.



(b) Selecting the best model from those trained with various batch sizes.

Figure 16. Selecting temperature using LMC and CKA similarity (squares) leads to a smaller test error than selecting temperature using the test error of the unpruned dense model (crosses). The retraining epochs of the proposed method $\alpha = 2$. The performance of CKA-based selection is close to the best test error found by grid search (dashed lines). (Left) Selecting the best training epoch. (Right) Selecting the best batch size. Models that perform significantly worse than grid search tend to have worse LMC, shown by the dark color of markers.

C.4. SAM hyperparameter ρ range

SAM was originally a method proposed to improve generalization in Foret et al. (2021); here we use it to train dense models to improve network pruning which is a different task. Thus, we would not expect our optimal ρ to be the same as the range in Foret et al. (2021). Furthermore, in our experiments, the range of ρ that we sweep over indeed includes 0.01, 0.02, 0.05, 0.1, 0.2, 0.5, and we find that using the ρ slightly larger than the original range provides more improvement on pruning.